Platform: Google Cloud BigQuery
Language: SQL
Dataset: MIMICIII
Data preprocessing: The goal is to find the relations and connection between files contained in MIMICIII dataset, do analysis and generate a CSV file that contains some patient's information related to a specific disease Diabetes (includes related lab results and patients' background information).

The project decided to collect patients' information and Diabetes related lab results.

--- Step 1: create a new table to record all subject_id that with diabetes diagnosis

```
create table `dark-rarity-400103.123456.subid` AS
SELECT
  p.subject_id
FROM
  physionet-data.mimiciii_clinical.patients AS p
INNER JOIN
  physionet-data.mimiciii_clinical.diagnoses_icd AS d
ON
  p.subject_id = d.subject_id
WHERE
  d.icd9_code LIKE '250%' -- ICD-9 code for diabetes
ORDER BY
  p.subject_id;
```

--- Step 2: create a new table that contains all primarily selected lab test and it's ID number

```
-- Create a new table named "data1"
CREATE TABLE `dark-rarity-400103.123456.data1`
AS
SELECT itemid as itemid_, label as label_
FROM `physionet-data.mimiciii_clinical.d_labitems`
WHERE label LIKE '%Hemoglobin%'
   OR label LIKE '%Glucose%'
   OR label LIKE '%Lactate%'
   OR label LIKE '%Sodium%'
   OR label LIKE '%Potassium%'
   OR label LIKE '%Chloride%'
   OR label LIKE '%Creatinine%'
   OR label LIKE '%Cholesterol%'
   OR label LIKE '%Urea Nitrogen%'
   OR label LIKE '%Triglycerides%'
   OR label LIKE '%pH%'
   OR label LIKE '%pCO2%';
```

```
--- Step: create a new table that contains all information (basic) and lab results
selected in previous steps of all patients

CREATE TABLE `dark-rarity-400103.123456.withDia`
AS
SELECT SUBJECT_ID,HADM_ID,ITEMID,VALUENUM,FLAG
FROM `physionet-data.mimiciii_clinical.labevents` le
WHERE le.itemid IN (SELECT itemid_ FROM `dark-rarity-400103.123456.data1`);

--- Step: counting the categories of lab test, deleting those that are not commonly
tested (< 2000) by patients

SELECT itemid, COUNT(*) as item_count FROM `physionet-
data.mimiciii_clinical.labevents`
GROUP BY itemid
ORDER BY item_count DESC;

-- Create a temporary table with the counts of each itemid in withDia
CREATE TEMP TABLE temp_counts AS (
  SELECT
    itemid,
    COUNT(*) AS itemid_count
  FROM
    `dark-rarity-400103.123456.withDia`
  GROUP BY
    itemid
);

-- Delete rows from data1 where itemid count is less than 2000 in withDia
DELETE FROM `dark-rarity-400103.123456.data1`
WHERE itemid_ IN (
  SELECT itemid
  FROM temp_counts
  WHERE itemid_count < 2000
);

-- Drop the temporary table
DROP TABLE temp_counts;

--- Step: create target variable, 0 means no diabetes, 1 means yes.

create table `dark-rarity-400103.123456.final` AS

SELECT
  withdia.*,
  CASE WHEN EXISTS (
    SELECT 1
    FROM `dark-rarity-400103.123456.subid` AS sub
```

```
    WHERE sub.subject_id = withdia.subject_id
  ) THEN 1 ELSE 0 END AS has_match
FROM
  `dark-rarity-400103.123456.withDiaAndPatient` AS withdia;

--- Step: counting the categories of lab test, deleting those that are not commonly
tested (< 2000) by patients again

-- Create a temporary table with the counts of each itemid in withDia
CREATE TEMP TABLE temp_counts AS (
  SELECT
    itemid,
    COUNT(*) AS itemid_count
  FROM
    `dark-rarity-400103.123456.final`
  GROUP BY
    itemid
);


-- Delete rows from data1 where itemid count is less than 2000 in final
DELETE FROM `dark-rarity-400103.123456.data1`
WHERE itemid_ IN (
  SELECT itemid
  FROM temp_counts
  WHERE itemid_count < 2000
);

-- Drop the temporary table
DROP TABLE temp_counts;

--- Step: make sure patients who were cleaned are all removed

-- Delete rows from final where itemid is not in data1
DELETE FROM `dark-rarity-400103.123456.final`
WHERE itemid NOT IN (
  SELECT itemid_
  FROM `dark-rarity-400103.123456.data1`
);

--- Step: cleaning column “HADM_ID” which won’t be used in project

ALTER TABLE `dark-rarity-400103.123456.final`
DROP COLUMN HADM_ID;

--- Step: collecting how many diabetes and non-diabetes

SELECT
```

```
  COUNT(DISTINCT subject_id) AS distinct_subjects_with_match
FROM
  `dark-rarity-400103.123456.final`
WHERE
  has_match = 0;

SELECT
  COUNT(DISTINCT subject_id) AS distinct_subjects_with_match
FROM
  `dark-rarity-400103.123456.final`
WHERE
  has_match = 1;

--- Step: cleaning subject_id who has less than 10 lab results

DELETE FROM `dark-rarity-400103.123456.final`
WHERE subject_id IN (
  SELECT subject_id
  FROM `dark-rarity-400103.123456.final`
  GROUP BY subject_id
  HAVING COUNT(*) < 10
);

--- Step: cleaning NULL data

DELETE FROM `dark-rarity-400103.123456.final`
WHERE VALUENUM IS NULL;

--- Step: for each patient if there's any repeat lab results, only keep the first
appearance

CREATE TABLE `dark-rarity-400103.123456.final_` AS
WITH RankedData AS (
  SELECT
    *,
    ROW_NUMBER() OVER (PARTITION BY subject_id, itemid) AS row_rank
  FROM
   `dark-rarity-400103.123456.final`
)
SELECT
  *,
FROM
  RankedData
WHERE
  row_rank = 1;

--- Step: ordered by subject_id
```

```sql
create table `dark-rarity-400103.123456.order_final` AS
select * from `dark-rarity-400103.123456.final_`
ORDER BY subject_id;
```